



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

executable model

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used executable model

Found 90,407 of 151,219

Sort results by
[Save results to a Binder](#)
[Try an Advanced Search](#)Display results
[Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** Executable JVM model for analytical reasoning: a study

Hanbing Liu, J Strother Moore

June 2003 **Proceedings of the 2003 workshop on Interpreters, Virtual Machines and Emulators**

Full text available: pdf(230.18 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

To study the properties of the Java Virtual Machine(JVM) and Java programs, our research group has produced a series of JVM models written in a functional subset of Common Lisp. In this paper, we present our most complete JVM model from this series, namely, M6, which is derived from a careful study of the J2ME KVM [16] implementation. On the one hand, our JVM model is a conventional machine emulator. M6 models accurately almost all aspects of the KVM implementation, including the dynamic class lo ...

2 An execution model for limited ambiguity rules and its application to derived data update

I.-Min A. Chen, Richard Hull, Dennis McLeod

December 1995 **ACM Transactions on Database Systems (TODS)**, Volume 20 Issue 4

Full text available: pdf(3.36 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A novel execution model for rule application in active databases is developed and applied to the problem of updating derived data in a database represented using a semantic, object-based database model. The execution model is based on the use of "limited ambiguity rules" (LARs), which permit disjunction in rule actions. The execution model essentially performs a breadth-first exploration of alternative extensions of a user-requested update. Given an object-based database schema, ...

Keywords: active database systems, deltas on database states, derived data, limited ambiguity rules, semantic data models, update propagation

3 A visual execution model for Ada tasking

Laura K. Dillon

October 1993 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 2 Issue 4

Full text available: pdf(2.08 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A visual execution model for Ada tasking can help programmers attain a deeper understanding of the tasking semantics. It can illustrate subtleties in semantic definitions that are not apparent in natural language design. We describe a contour model of Ada tasking that depicts asynchronous tasks (threads of control), relationships between the environments in which tasks execute, and the manner in which tasks interact. The use of this high-level execution model makes it possible to see what h ...

Keywords: contour model, visual execution model

4 Analysis of Or-parallel execution models

Gopal Gupta, Bharat Jayaraman

September 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 15 Issue 4

Full text available:  pdf(1.62 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We discuss fundamental limitations of or-parallel execution models of nondeterministic programming languages. Or-parallelism corresponds to the execution of different nondeterministic computational paths in parallel. A natural way to represent the state of (parallel) execution of a nondeterministic program is by means of an or-parallel tree. We identify three important criteria that underlie the design of or-parallel implementations based on the or-parallel tree: constant-time access to var ...

Keywords: Or-parallel execution models

5 Direct execution models of processor behavior and performance

Richard M. Fujimoto, William B. Campbell

December 1987 **Proceedings of the 19th conference on Winter simulation**

Full text available:  pdf(952.49 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper discusses a modeling technique for creating efficient instruction level simulation models of von Neumann processors. In contrast to traditional approaches which use a software interpreter, this technique employs direct execution of application programs on the host computer. An assembly language program for the target machine is decompiled to a high level language, instrumented, and then recompiled and executed on the host computer. A prototype im ...

6 Speculative execution model with duplication

Kei Hiraki, Junji Tamatsukuri, Takashi Matsumoto

July 1998 **Proceedings of the 12th international conference on Supercomputing**

Full text available:  pdf(988.00 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)

7 A flexible operation execution model for shared distributed objects

Saniya Ben Hassen, Irina Athanasiu, Henri E. Bal

October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 31 Issue 10

Full text available:  pdf(2.30 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many parallel and distributed programming models are based on some form of shared objects, which may be represented in various ways (e.g., single-copy, replicated, and


partitioned objects). Also, many different operation *execution strategies* have been designed for each representation. In programming systems that use multiple representations integrated in a single object model, one way to provide multiple execution strategies is to implement each strategy independently from the others. How ...

8 A hybrid execution model for fine-grained languages on distributed memory multicomputers

John Plevyak, Vijay Karamcheti, Xingbin Zhang, Andrew A. Chien

December 1995 **Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available:  [html\(59.68 KB\)](#)


 [ps\(419.70 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 OFL: a functional execution model for object query languages

Georges Gardarin, Fernando Machuca, Philippe Pucheral

May 1995 **ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data**, Volume 24 Issue 2

Full text available:  [pdf\(1.40 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a functional paradigm for querying efficiently abstract collections of complex objects. Abstract collections are used to model class extents, multivalued attributes as well as indexes or hashing tables. Our paradigm includes a functional language called OFL (Object Functional Language) and a supporting execution model based on graph traversals. OFL is able to support any complex object algebra with recursion as macros. It is an appropriate target language for OQL-like query compilers. ...

10 A model for dataflow based vector execution

W. Marcus Miller, Walid A. Najjar, A. P. Wim Böhm

July 1994 **Proceedings of the 8th international conference on Supercomputing**

Full text available:  [pdf\(1.28 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Although the dataflow model has been shown to allow the exploitation of parallelism at all levels, research of the past decade has revealed several fundamental problems: Synchronization at the instruction level, token matching, coloring and re-labeling operations have a negative impact on performance by significantly increasing the number of non-compute "overhead" cycles. Recently, many novel Hybrid von-Neumann Data Driven machines have been proposed to alleviate some of these p ...

11 Sentinel scheduling: a model for compiler-controlled speculative execution

Scott A. Mahlke, William Y. Chen, Roger A. Bringmann, Richard E. Hank, Wen-Mei W. Hwu, B. Ramakrishna Rau, Michael S. Schlansker

November 1993 **ACM Transactions on Computer Systems (TOCS)**, Volume 11 Issue 4

Full text available:  [pdf\(2.26 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Speculative execution is an important source of parallelism for VLIW and superscalar processors. A serious challenge with compiler-controlled speculative execution is to efficiently handle exceptions for speculative instructions. In this article, a set of architectural features and compile-time scheduling support collectively referred to as sentinel scheduling is introduced. Sentinel scheduling provides an effective framework for both compiler-controlled speculative executi ...

Keywords: VLIW processor, exception detection, exception recovery, instruction scheduling,

instruction-level parallelism, speculative execution, superscalar processor

12 Software process modeling and execution within virtual environments

John C. Doppke, Dennis Heimbigner, Alexander L. Wolf

January 1998 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 7 Issue 1

Full text available:  pdf(232.51 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In the past, multiuser virtual environments have been developed as venues for entertainment and social interaction. Recent research focuses instead on their utility in carrying out work in the real world. This research has identified the importance of a mapping between the real and the virtual that permits the representation of real tasks in the virtual environment. We investigate the use of virtual environments—in particular, MUDs (Multi-User Dimensions)—in the domain of softwa ...

Keywords: MOO, MUD, PROMO, software process, tools, virtual environments

13 A heuristics-based approach to decreasing simulation model execution cost

Keith B. Johnston, Jan A. Aitken

March 1983 **Proceedings of the 16th annual symposium on Simulation**

Full text available:  pdf(426.77 KB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a brief overview of existing techniques oriented toward improving the executional efficiency of simulation models. It then presents a new heuristics-based technique, describing some language, data, and algorithmic requirements for a simulation package to selectively remove levels of detail from a model in such a way as to decrease the execution time requirements (and cost) of experimenting with the model.

14 An empirical study of decentralized ILP execution models

Narayan Ranganathan, Manoj Franklin

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32
Issue 11 , 5

Full text available:  pdf(1.25 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Recent fascination for dynamic scheduling as a means for exploiting instruction-level parallelism has introduced significant interest in the scalability aspects of dynamic scheduling hardware. In order to overcome the scalability problems of centralized hardware schedulers, many decentralized execution models are being proposed and investigated recently. The crux of all these models is to split the instruction window across multiple processing elements (PEs) that do independent, scheduling of in ...

Keywords: control dependence, data dependence, decentralization, dynamic scheduling, execution unit dependence, hardware window, instruction-level parallelism, speculative execution

15 Safely executing untrusted code: Model-carrying code: a practical approach for safe execution of untrusted applications

R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. DuVarney

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Full text available:  [pdf\(301.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a new approach called *model-carrying code* (MCC) for safe execution of untrusted code. At the heart of MCC is the idea that untrusted code comes equipped with a concise high-level model of its security-relevant behavior. This model helps bridge the gap between high-level security policies and low-level binary code, thereby enabling analyses which would otherwise be impractical. For instance, users can use a fully automated verification procedure to determine if the code ...

Keywords: mobile code security, policy enforcement, sand-boxing, security policies

16 Synthesis for Low Power: Dynamic modeling of inter-instruction effects for execution time estimation

G. Beltrame, C. Brandolese, W. Fornaciari, F. Salice, D. Sciuto, V. Trianni

September 2001 **Proceedings of the 14th international symposium on Systems synthesis**

Full text available:  [pdf\(234.84 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The market for embedded applications is facing a growing interest in power consumption issues: this work is intended to provide a new model to estimate software-level power consumption of 32-bit microprocessors. This model extends previous ones by considering dynamic inter-instruction effects that take place during code execution, providing a static means to characterize their energy consumption. The model is formally sound: it is conceived for a generic architecture and it has been preliminary ...

17 Towards demystification of direct manipulation: cognitive modeling charts the gulf of execution

David Kieras, David Meyer, James Ballas

March 2001 **Proceedings of the SIGCHI conference on Human factors in computing systems**

Full text available:  [pdf\(147.38 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Direct manipulation involves a large number of interacting psychological mechanisms that make the performance of a given interface hard to predict on intuitive or informal grounds. This paper applies cognitive modeling to explain the subtle effects produced by using a keypad versus a touchscreen in a performance-critical laboratory task.

Keywords: cognitive modeling, direct manipulation

18 Session 4B: System level power and performance modeling: An assembly-level execution-time model for pipelined architectures

G. Beltrame, C. Brandolese, W. Fornaciari, F. Salice, D. Sciuto, V. Trianni

November 2001 **Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design**

Full text available:  [pdf\(117.63 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The aim of this work is to provide an elegant and accurate static execution timing model for 32-bit microprocessor instruction sets, covering also inter-instruction effects. Such effects depend on the processor state and the pipeline behavior, and are related to the dynamic execution of assembly code. The paper proposes a mathematical model of the delays deriving from instruction dependencies and gives a statistical characterization of such timing overheads. The model has been validated on a com ...

19 A parallel execution model for a database machine with high performances

Didier Donsez, Pascal Faudemay

July 1990 **Proceedings of the second international symposium on Databases in parallel and distributed systems**Full text available:  [pdf\(1.47 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we present a mixed MIMD / SIMD execution model for a reconfigurable computer. This model is adapted to the use of a specialized associative coprocessor, embedded in this host machine. A main characteristic of the model is that it uses four types of processes (decoding, calculus, coprocessor communication and transaction manager), and that in principle one process of each type is allowed on each processor. Time intervals are allocated to operations into partitions of t ...

20 Modeling methodology: Techniques for optimizing model execution I:**aggressiveness/risk effects based scheduling in Time Warp**

Vittorio Cortellessa, Francesco Quaglia

December 2000 **Proceedings of the 32nd conference on Winter simulation**Full text available:  [pdf\(94.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

The Time Warp synchronization protocol for parallel discrete event simulation is characterized by aggressiveness and risk. The former property refers to greediness in the execution of unsafe events. The latter one refers to greediness in the notification of new events produced by aggressive event execution. Both these properties are potential sources for rollback occurrence/spreading. In this paper we present a scheduling algorithm for the selection of the next LP to be run on a processor which te ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **executable modeling language**

 Found **74,472** of **151,219**

 Sort results by
☒ [Save results to a Binder](#)
[Try an Advanced Search](#)

 Display results
☒ [Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [OFL: a functional execution model for object query languages](#)

Georges Gardarin, Fernando Machuca, Philippe Pucheral

 May 1995 **ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data**, Volume 24 Issue 2

 Full text available: [pdf\(1.40 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a functional paradigm for querying efficiently abstract collections of complex objects. Abstract collections are used to model class extents, multivalued attributes as well as indexes or hashing tables. Our paradigm includes a functional language called OFL (Object Functional Language) and a supporting execution model based on graph traversals. OFL is able to support any complex object algebra with recursion as macros. It is an appropriate target language for OQL-like query compilers. ...

2 [A hybrid execution model for fine-grained languages on distributed memory multicomputers](#)

John Plevyak, Vijay Karamcheti, Xingbin Zhang, Andrew A. Chien

 December 1995 **Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)**

 Full text available: [html\(59.68 KB\)](#)
[ps\(419.70 KB\)](#)

 Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

3 [Analysis of Or-parallel execution models](#)

Gopal Gupta, Bharat Jayaraman

 September 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 4

 Full text available: [pdf\(1.62 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We discuss fundamental limitations of or-parallel execution models of nondeterministic programming languages. Or-parallelism corresponds to the execution of different nondeterministic computational paths in parallel. A natural way to represent the state of (parallel) execution of a nondeterministic program is by means of an or-parallel tree. We identify three important criteria that underlie the design of or-parallel implementations based on the or-parallel tree: constant-time access to var ...

Keywords: Or-parallel execution models

4 A visual execution model for Ada tasking

Laura K. Dillon

October 1993 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 2 Issue 4

Full text available:  pdf(2.08 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A visual execution model for Ada tasking can help programmers attain a deeper understanding of the tasking semantics. It can illustrate subtleties in semantic definitions that are not apparent in natural language design. We describe a contour model of Ada tasking that depicts asynchronous tasks (threads of control), relationships between the environments in which tasks execute, and the manner in which tasks interact. The use of this high-level execution model makes it possible to see what h ...

Keywords: contour model, visual execution model

5 Models and languages for parallel computation

David B. Skillicorn, Domenico Talia

June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2

Full text available:  pdf(298.05 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We survey parallel programming models and languages using six criteria to assess their suitability for realistic portable parallel programming. We argue that an ideal model should be easy to program, should have a software development methodology, should be architecture-independent, should be easy to understand, should guarantee performance, and should provide accurate information about the cost of programs. These criteria reflect our belief that developments in parallelism must be driven b ...

Keywords: general-purpose parallel computation, logic programming languages, object-oriented languages, parallel programming languages, parallel programming models, software development methods, taxonomy

6 Advances in system specification and system design frameworks: Concurrent execution semantics and sequential simulation algorithms for the metropolis meta-model

Felice Balarin, Luciano Lavagno, Claudio Passerone, Alberto Sangiovanni-Vincentelli, Yosinori Watanabe, Guang Yang

May 2002 **Proceedings of the tenth international symposium on Hardware/software codesign**

Full text available:  pdf(596.18 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents the simulation techniques that are available in Metropolis, an interdisciplinary research project that develops a design methodology, supported by a comprehensive design environment and tool set, for embedded systems. System behavior is non-deterministic in general, especially in the beginning of the design process, when several key decision, such as the mapping on an implementation platform, have not yet been made, and thus the traces obtainable by simulation are not unique ...

7 Executable object modeling with statecharts

David Harel, Eran Gery

May 1996 **Proceedings of the 18th international conference on Software engineering**

Full text available:  pdf(1.43 MB)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

This paper reports on an effort to develop an integrated set of diagrammatic languages for modeling object-oriented systems, and to construct a supporting tool. The goal is for models to be intuitive and well-structured, yet fully executable and analyzable, enabling automatic synthesis of usable and efficient code in object-oriented languages such as C++. At the heart of the modeling method is the language of statecharts for specifying object behavior, and a hierarchical OMT-like language for de ...

Keywords: C++, O-charts, active objects, diagrammatic languages, direct invocation, executable object modeling, hierarchical OMT-like language, integrated set, multiple-thread concurrency, object-oriented languages, object-oriented programming, object-oriented systems, software tools, statecharts

8 [Sequential description and parallel execution language DFCII dataflow supercomputers](#)

Satoshi Sekiguchi, Toshio Shimada, Kei Hiraki

June 1991 **Proceedings of the 5th international conference on Supercomputing**

Full text available:  pdf(793.78 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 [μ3L: An HLL-RISC processor for parallel execution of FP-language programs](#)

M. Castan, E. I. Organick

April 1982 **Proceedings of the 9th annual symposium on Computer Architecture**

Full text available:  pdf(709.88 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

To eliminate the conceptual distance between the hardware instruction set and the user interface, some architects advocate High Level Language (HLL) machines. To obtain simple, fast and cheap machines, some architects advocate Reduced Instruction Set Computer (RISC) machines. This paper reconciles both views and presents an architecture which has both an HLL user interface and a RISC hardware. Each instance of this architecture is a module of an HLL multiprocessor system. Functio ...

10 [A flexible operation execution model for shared distributed objects](#)

Saniya Ben Hassen, Irina Athanasias, Henri E. Bal

October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 31 Issue 10

Full text available:  pdf(2.30 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many parallel and distributed programming models are based on some form of shared objects, which may be represented in various ways (e.g., single-copy, replicated, and partitioned objects). Also, many different operation *execution strategies* have been designed for each representation. In programming systems that use multiple representations integrated in a single object model, one way to provide multiple execution strategies is to implement each strategy independently from the others. How ...

11 [Modeling languages versus matrix generators for linear programming](#)

Robert Fourer

June 1983 **ACM Transactions on Mathematical Software (TOMS)**, Volume 9 Issue 2

Full text available:  pdf(2.86 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 Linking simulation model specification and parallel execution through UNITY

Marc Abrams, Ernest H. Page, Richard E. Nance

December 1991 **Proceedings of the 23rd conference on Winter simulation**


Full text available:  pdf(918.07 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



13 A specification language to assist in analysis of discrete event simulation models

C Michael Overstreet, Richard E. Nance

February 1985 **Communications of the ACM**, Volume 28 Issue 2

Full text available:  pdf(1.37 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)




Effective development environments for discrete event simulation models should reduce development costs and improve model performance. A model specification language used in a model development environment is defined. This approach is intended to reduce modeling costs by interposing an intermediate form between a conceptual model (the model as it exists in the mind of the modeler) and an executable representation of that model. As a model specification is constructed, the incomplete specifi ...

14 A dynamically configurable, multi-language execution platform

Bertil Folliot, Ian Piumarta, Fabio Riccardi

September 1998 **Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications**

Full text available:  pdf(449.77 KB) Additional Information: [full citation](#), [citations](#), [index terms](#)



15 Processor System Modeling - a language and simulation system

Fredrick Pfisterer, William C. Purdy, Leonard Lehmann

July 1976 **ACM SIGSIM Simulation Digest , Proceedings of the 4th symposium on Simulation of computer systems**, Volume 7 Issue 4

Full text available:  pdf(538.48 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)



There is a need for a language and simulation system that computer systems designers (rather than programmers) can use to build and exercise models of computer systems. The major characteristics of such a language and system are presented. A brief overview of the Processor System Modeling (PSM) system, consisting of PSM language input, translation to Simscript, execution, and interactive post processor operation on a trace of the results, is outlined. The different typ ...

16 An execution model for limited ambiguity rules and its application to derived data update

I.-Min A. Chen, Richard Hull, Dennis McLeod

December 1995 **ACM Transactions on Database Systems (TODS)**, Volume 20 Issue 4

Full text available:  pdf(3.36 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)



A novel execution model for rule application in active databases is developed and applied to the problem of updating derived data in a database represented using a semantic, object-based database model. The execution model is based on the use of "limited ambiguity rules" (LARs), which permit disjunction in rule actions. The execution model essentially performs a breadth-first exploration of alternative extensions of a user-requested update. Given an object-based database schema, ...

Keywords: active database systems, deltas on database states, derived data, limited ambiguity rules, semantic data models, update propagation

17 XML manipulations: Experimenting with the circus language for XML modeling and transformation

Jean-Yves Vion-Dury, Veronika Lux, Emmanuel Pietriga

November 2002 **Proceedings of the 2002 ACM symposium on Document engineering**

Full text available:  pdf(81.56 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

After a brief introduction to the Circus programming language, we present a simple type set to model XML structures. We then describe a transformation that takes a mail as input and produces a reply, showing how subtyping is used in order to refine the type control and specialize the transformation. Conclusions are drawn both on our (easy to use but clearly limited) XML data model and on Circus itself ; expected qualities of the language are verified ; the need for some new features is expressed ...

Keywords: XML, XSLT, circus, document model, programming language, typed document transformation

18 Executable JVM model for analytical reasoning: a study

Hanbing Liu, J Strother Moore

June 2003 **Proceedings of the 2003 workshop on Interpreters, Virtual Machines and Emulators**

Full text available:  pdf(230.18 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

To study the properties of the Java Virtual Machine(JVM) and Java programs, our research group has produced a series of JVM models written in a functional subset of Common Lisp. In this paper, we present our most complete JVM model from this series, namely, M6, which is derived from a careful study of the J2ME KVM [16] implementation. On the one hand, our JVM model is a conventional machine emulator. M6 models accurately almost all aspects of the KVM implementation, including the dynamic class lo ...

19 Technical correspondence: Method driven model: a unified model for an object composition language

Chitra Babu, D. Janakiram

August 2004 **ACM SIGPLAN Notices**, Volume 39 Issue 8

Full text available:  pdf(199.59 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Object-Oriented (OO) paradigm facilitates identification of pertinent domain objects as nouns. Often the domain verbs are modeled as member functions associated with these objects. However, object interactions are also characterized by domain verbs. The current OO programming languages lack suitable abstractions for modeling interactions among objects. To overcome this limitation, we propose a unified model known as Method Driven Model (MDM) that employs a novel approach for capturing object ...


Keywords: Object Schizophrenia Problem, aspect, object identity

20 Direct execution models of processor behavior and performance

Richard M. Fujimoto, William B. Campbell

December 1987 **Proceedings of the 19th conference on Winter simulation**

Full text available: Additional Information:

 [pdf\(952.49 KB\)](#)[full citation](#), [abstract](#), [references](#), [index terms](#)

This paper discusses a modeling technique for creating efficient instruction level simulation models of von Neumann processors. In contrast to traditional approaches which use a software interpreter, this technique employs direct execution of application programs on the host computer. An assembly language program for the target machine is decompiled to a high level language, instrumented, and then recompiled and executed on the host computer. A prototype im ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)